Lstm english to hindi [seq to seq model]

- data_source:http://www.cfilt.iitb.ac.in/iitb_parallel/iitb_corpus_download/?C=D;O=A (http://www.cfilt.iitb.ac.in/iitb_parallel/iitb_corpus_download/?C=D;O=A)

In [1]:

```
import string
import re
from numpy import array, argmax, random, take
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense, LSTM, Embedding, RepeatVector
from keras.preprocessing.text import Tokenizer
from keras.callbacks import ModelCheckpoint
from keras.preprocessing.sequence import pad_sequences
from keras.models import load_model
from keras import optimizers
import matplotlib.pyplot as plt
%matplotlib inline
#pd.set_option('display.max_colwidth', 200)
```

```
Using TensorFlow backend.
```

In [2]:

```
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
#https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/
# function to read raw text file
def read_text(filename):
        # open the file
        file = open(filename, mode='rt', encoding='utf-8')

        # read all text
        text = file.read()
        file.close()
        return text
```

In [4]:

```
#https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/
# split a text into sentences
def to_lines(text):
    sents = text.strip().split('\n')
    #sents = [i.split('\t') for i in sents]
    return sents
```

In [5]:

```
#data_source:http://www.cfilt.iitb.ac.in/iitb_parallel/iitb_corpus_download/?C
=D;O=A
text_hin=read_text('parallel_hin_eng/parallel/IITB.en-hi.hi')
text_eng=read_text('parallel_hin_eng/parallel/IITB.en-hi.en')
data_hin=to_lines(text_hin)
data_eng=to_lines(text_eng)
```

In [6]:

```
#converting sentences to dataframe
df_hin=pd.DataFrame(data_hin)
df_eng=pd.DataFrame(data_eng)
```

In [7]:

```
df_eng.shape,df_hin.shape
```

Out[7]:

```
((1561840, 1), (1561840, 1))
```

In [8]:

```
#removing punctutations from dataframe
#https://stackoverflow.com/questions/39782418/remove-punctuations-in-pandas
df_eng[0]=df_eng[0].str.replace('[{}]'.format(string.punctuation), '')
df_hin[0]=df_hin[0].str.replace('[{}]'.format(string.punctuation), '')
df_eng.shape==df_hin.shape
```

Out[8]:

```
True
```

In [9]:

```
#converting all dataframe to lowercase
df_eng[0]=df_eng[0].str.lower()
df_hin[0]=df_hin[0].str.lower()
```

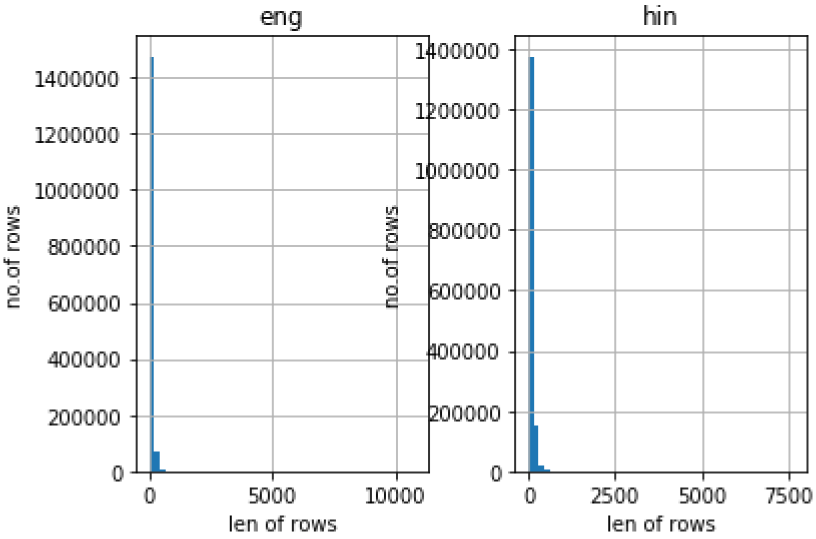
In [10]:

```
#ref:https://stackoverflow.com/questions/29314033/drop-rows-containing-empty-c
ells-from-a-pandas-dataframe
eng_zero=[]
hin_zero=[]
c=0
for i in df_eng[0].str.strip().astype(bool):
    if(i==False):
        eng_zero.append(c)
    c+=1
    pass
c=0
for i in df_hin[0].str.strip().astype(bool):
    if(i==False):
        hin_zero.append(c)
    c+=1
    pass
remove_list=list(set(eng_zero+hin_zero))#remove_list has rows of empty values

#removing empty rows
df_eng.drop(df_eng.index[remove_list], inplace=True)
df_eng.reset_index(inplace=True)
df_hin.drop(df_hin.index[remove_list], inplace=True)
df_hin.reset_index(inplace=True)
```

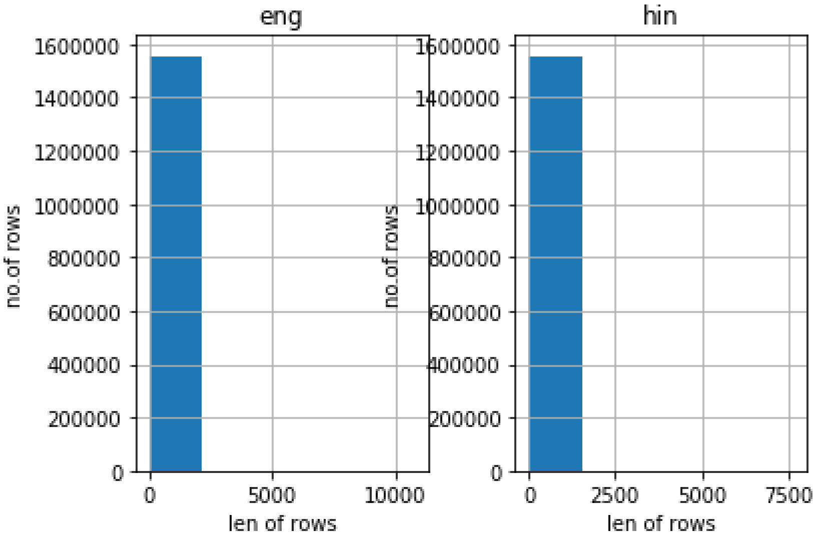
In [11]:
```
#EDA
df_len=pd.DataFrame()
df_len['eng']=df_eng[0].apply(len)
df_len['hin']=df_hin[0].apply(len)
fig=df_len.hist(bins =50)

for ax in fig.flatten():
    ax.set_xlabel("len of rows")
    ax.set_ylabel("no.of rows")
```



In [12]:
```
fig=df_len.hist(bins =5)
for ax in fig.flatten():
    ax.set_xlabel("len of rows")
    ax.set_ylabel("no.of rows")
    pass
print('For english sentences *max_len=',df_len['eng'].max(),'*min_len=',df_len
['eng'].min(),'*avg_len=',df_len['eng'].mean())
print('For hindi sentences *max_len=',df_len['hin'].max(),'*min_len=',df_len[
'hin'].min(),'*avg_len=',df_len['hin'].mean())
```

```
For english sentences *max_len= 10778 *min_len= 1 *avg_len= 70.60703200139896
For hindi sentences *max_len= 7645 *min_len= 1 *avg_len= 69.7714416316496
```



In [13]:
```
df_eng.head()
```

Out[13]:

	index	0
0	0	give your application an accessibility workout
1	1	accerciser accessibility explorer
2	2	the default plugin layout for the bottom panel
3	3	the default plugin layout for the top panel
4	4	a list of plugins that are disabled by default

In [14]:
```
df_hin.head()
```

Out[14]:

	index	0
0	0	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें
1	1	एक्सेर्साइसर पहुंचनीयता अन्वेषक
2	2	निचले पटल के लिए डिफोल्ट प्लगइन खाका
3	3	ऊपरी पटल के लिए डिफोल्ट प्लगइन खाका
4	4	उन प्लगइनों की सूची जिन्हें डिफोल्ट रूप से निष...

In [15]:

```
#ref:=http://carrefax.com/new-blog/2018/9/5/find-rows-containing-specific-values-in-a-pandas-dataframe
eng_zero=[]
hin_zero=[]
c=0
for i in df_eng[0].str.contains('([a-z])'):
    if(i==False):
        eng_zero.append(c)
    c+=1
    pass
c=0
for i in df_hin[0].str.contains('([a-z])'):
    if(i==True):
        hin_zero.append(c)
    c+=1
    pass
remove_list=list(set(eng_zero+hin_zero))#remove rows which contain eng words in hindhi sentences and vice versa

#removing rows
df_eng.drop(df_eng.index[remove_list], inplace=True)
df_eng.reset_index(inplace=True)

df_hin.drop(df_hin.index[remove_list], inplace=True)
df_hin.reset_index(inplace=True)

print("percentage of data points removed =",(len(remove_list)/len(df_eng[0]))*100)
```

```
percentage of data points removed = 9.178406227407049
```

In [16]:

```
#ref:https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/
#https://machinelearningmastery.com/develop-neural-machine-translation-system-keras/
#https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa
def tokenization(lines):
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(lines)
    return tokenizer
```

In [58]:

```
# prepare english tokenizer
eng_tokenizer = tokenization(df_eng[0][:50000])
eng_vocab_size = len(eng_tokenizer.word_index) + 1

eng_length = 8
print('Size of English Vocabulary : %d' % eng_vocab_size)
```

```
Size of English Vocabulary : 3464
```

In [59]:
```
# prepare Deutch tokenizer
hin_tokenizer = tokenization(df_hin[0][:50000])
hin_vocab_size = len(hin_tokenizer.word_index) + 1

hin_length = 8
print('Size of Hindi Vocabulary : %d' % hin_vocab_size)
```

```
Size of Hindi Vocabulary : 4149
```

In [20]:
```
# encode and pad sequences
def encode_sequences(tokenizer, length, lines):
    # integer encode sequences
    seq = tokenizer.texts_to_sequences(lines)
    # pad sequences with 0 values
    seq = pad_sequences(seq, maxlen=length, padding='post')
    return seq
```

Model building

In [21]:
```
df=pd.DataFrame()
df['eng']=df_eng[0][:50000]
df['hin']=df_hin[0][:50000]
```

In [22]:
```
df.head(15)
```

Out[22]:

	eng	hin
0	give your application an accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें
1	accerciser accessibility explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक
2	the default plugin layout for the bottom panel	निचले पटल के लिए डिफोल्ट प्लगइन खाका
3	the default plugin layout for the top panel	ऊपरी पटल के लिए डिफोल्ट प्लगइन खाका
4	a list of plugins that are disabled by default	उन प्लगइनों की सूची जिन्हें डिफोल्ट रूप से निष...
5	highlight duration	अवधि को हाइलाइट रकें
6	the duration of the highlight box when selecti...	पहुंचनीय आसंधि नोड को चुनते समय हाइलाइट बक्से ...
7	highlight border color	सीमांत बोर्डर के रंग को हाइलाइट करें
8	the color and opacity of the highlight border	हाइलाइट किए गए सीमांत का रंग और अपारदर्शिता।
9	highlight fill color	भराई के रंग को हाइलाइट करें
10	the color and opacity of the highlight fill	हाइलाइट किया गया भराई का रंग और पारदर्शिता।
11	api browser	एपीआई विचरक
12	browse the various methods of the current acce...	इस समय जिसे प्राप्त किया गया हो उसकी विभिन्न व...
13	hide private attributes	निजी गुणों को छिपाएं
14	method	विधि

In [25]:
```
#SPLITTING DATA
from sklearn.model_selection import train_test_split
train, test = train_test_split(df, test_size=0.25, random_state = 42)
# prepare training data
trainX = encode_sequences(eng_tokenizer, eng_length, train.iloc[:,0])
trainY = encode_sequences(hin_tokenizer, hin_length, train.iloc[:,1])
# prepare validation data
testX = encode_sequences(eng_tokenizer, eng_length, test.iloc[:,0])
testY = encode_sequences(hin_tokenizer, hin_length, test.iloc[:,0])
```

In [27]:

```
#model
#ref:https://keras.io/layers/core/
#ref:https://datascience.stackexchange.com/questions/46491/what-is-the-job-of-repeatvector-and-timedistributed
model = Sequential()
model.add(Embedding(eng_vocab_size, 512, input_length=eng_length, mask_zero=True))
model.add(LSTM(512))
model.add(RepeatVector(hin_length))
model.add(LSTM(512, return_sequences=True))
model.add(Dense(hin_vocab_size, activation='softmax'))
```

```
WARNING:tensorflow:From F:\anaconda3\lib\site-packages\tensorflow\python\fram
ework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.
ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```

In [28]:

```
#https://machinelearningmastery.com/develop-neural-machine-translation-system-keras/
rms = optimizers.RMSprop(lr=0.001)
model.compile(optimizer=rms, loss='sparse_categorical_crossentropy')
```

In [29]:

```
#https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa#saving model
filename = 'model_eng2hin.h1.24_jan_19'
checkpoint = ModelCheckpoint(filename, monitor='val_loss', verbose=1, save_best_only=True, mode='min')

history = model.fit(trainX, trainY.reshape(trainY.shape[0], trainY.shape[1], 1),
          epochs=50, batch_size=256,
          validation_split = 0.25,
          callbacks=[checkpoint], verbose=1)
```

```
WARNING:tensorflow:From F:\anaconda3\lib\site-packages\tensorflow\python\ops
\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecat
ed and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From F:\anaconda3\lib\site-packages\tensorflow\python\ops
\math_grad.py:102: div (from tensorflow.python.ops.math_ops) is deprecated an
d will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Train on 28125 samples, validate on 9375 samples
Epoch 1/50
28125/28125 [==============================] - 176s 6ms/step - loss: 3.3790 -
val_loss: 2.9463

Epoch 00001: val_loss improved from inf to 2.94635, saving model to model_eng
2hin.h1.24_jan_19
Epoch 2/50
28125/28125 [==============================] - 176s 6ms/step - loss: 2.8377 -
val_loss: 2.7242

Epoch 00002: val_loss improved from 2.94635 to 2.72421, saving model to model
_eng2hin.h1.24_jan_19
Epoch 3/50
28125/28125 [==============================] - 173s 6ms/step - loss: 2.6454 -
val_loss: 2.5334

Epoch 00003: val_loss improved from 2.72421 to 2.53343, saving model to model
_eng2hin.h1.24_jan_19
Epoch 4/50
28125/28125 [==============================] - 172s 6ms/step - loss: 2.4410 -
val_loss: 2.4443

Epoch 00004: val_loss improved from 2.53343 to 2.44430, saving model to model
_eng2hin.h1.24_jan_19
Epoch 5/50
28125/28125 [==============================] - 169s 6ms/step - loss: 2.2233 -
val_loss: 2.2100

Epoch 00005: val_loss improved from 2.44430 to 2.20997, saving model to model
_eng2hin.h1.24_jan_19
Epoch 6/50
28125/28125 [==============================] - 168s 6ms/step - loss: 1.9992 -
val_loss: 1.9504

Epoch 00006: val_loss improved from 2.20997 to 1.95043, saving model to model
_eng2hin.h1.24_jan_19
Epoch 7/50
28125/28125 [==============================] - 168s 6ms/step - loss: 1.7722 -
val_loss: 1.7556

Epoch 00007: val_loss improved from 1.95043 to 1.75562, saving model to model
_eng2hin.h1.24_jan_19
Epoch 8/50
28125/28125 [==============================] - 168s 6ms/step - loss: 1.5508 -
val_loss: 1.5857
```

```
Epoch 00008: val_loss improved from 1.75562 to 1.58570, saving model to model
_eng2hin.h1.24_jan_19
Epoch 9/50
28125/28125 [==============================] - 169s 6ms/step - loss: 1.3440 -
val_loss: 1.3851

Epoch 00009: val_loss improved from 1.58570 to 1.38511, saving model to model
_eng2hin.h1.24_jan_19
Epoch 10/50
28125/28125 [==============================] - 169s 6ms/step - loss: 1.1534 -
val_loss: 1.2598

Epoch 00010: val_loss improved from 1.38511 to 1.25982, saving model to model
_eng2hin.h1.24_jan_19
Epoch 11/50
28125/28125 [==============================] - 170s 6ms/step - loss: 0.9857 -
val_loss: 1.0904

Epoch 00011: val_loss improved from 1.25982 to 1.09043, saving model to model
_eng2hin.h1.24_jan_19
Epoch 12/50
28125/28125 [==============================] - 169s 6ms/step - loss: 0.8360 -
val_loss: 0.9549

Epoch 00012: val_loss improved from 1.09043 to 0.95485, saving model to model
_eng2hin.h1.24_jan_19
Epoch 13/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.7114 -
val_loss: 0.8580

Epoch 00013: val_loss improved from 0.95485 to 0.85800, saving model to model
_eng2hin.h1.24_jan_19
Epoch 14/50
28125/28125 [==============================] - 168s 6ms/step - loss: 0.6020 -
val_loss: 0.7601

Epoch 00014: val_loss improved from 0.85800 to 0.76015, saving model to model
_eng2hin.h1.24_jan_19
Epoch 15/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.5118 -
val_loss: 0.6890

Epoch 00015: val_loss improved from 0.76015 to 0.68901, saving model to model
_eng2hin.h1.24_jan_19
Epoch 16/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.4343 -
val_loss: 0.6112

Epoch 00016: val_loss improved from 0.68901 to 0.61125, saving model to model
_eng2hin.h1.24_jan_19
Epoch 17/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.3708 -
val_loss: 0.5635

Epoch 00017: val_loss improved from 0.61125 to 0.56350, saving model to model
_eng2hin.h1.24_jan_19
Epoch 18/50
```

```
28125/28125 [==============================] - 168s 6ms/step - loss: 0.3203 -
val_loss: 0.5502

Epoch 00018: val_loss improved from 0.56350 to 0.55023, saving model to model
_eng2hin.h1.24_jan_19
Epoch 19/50
28125/28125 [==============================] - 168s 6ms/step - loss: 0.2766 -
val_loss: 0.4831

Epoch 00019: val_loss improved from 0.55023 to 0.48310, saving model to model
_eng2hin.h1.24_jan_19
Epoch 20/50
28125/28125 [==============================] - 168s 6ms/step - loss: 0.2434 -
val_loss: 0.4526

Epoch 00020: val_loss improved from 0.48310 to 0.45263, saving model to model
_eng2hin.h1.24_jan_19
Epoch 21/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.2162 -
val_loss: 0.4434

Epoch 00021: val_loss improved from 0.45263 to 0.44344, saving model to model
_eng2hin.h1.24_jan_19
Epoch 22/50
28125/28125 [==============================] - 168s 6ms/step - loss: 0.1935 -
val_loss: 0.4162

Epoch 00022: val_loss improved from 0.44344 to 0.41619, saving model to model
_eng2hin.h1.24_jan_19
Epoch 23/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1755 -
val_loss: 0.3973

Epoch 00023: val_loss improved from 0.41619 to 0.39734, saving model to model
_eng2hin.h1.24_jan_19
Epoch 24/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1600 -
val_loss: 0.3858

Epoch 00024: val_loss improved from 0.39734 to 0.38579, saving model to model
_eng2hin.h1.24_jan_19
Epoch 25/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1482 -
val_loss: 0.3769

Epoch 00025: val_loss improved from 0.38579 to 0.37690, saving model to model
_eng2hin.h1.24_jan_19
Epoch 26/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1375 -
val_loss: 0.3701

Epoch 00026: val_loss improved from 0.37690 to 0.37009, saving model to model
_eng2hin.h1.24_jan_19
Epoch 27/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1295 -
val_loss: 0.3646
```

```
Epoch 00027: val_loss improved from 0.37009 to 0.36455, saving model to model
_eng2hin.h1.24_jan_19
Epoch 28/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1225 -
val_loss: 0.3601

Epoch 00028: val_loss improved from 0.36455 to 0.36006, saving model to model
_eng2hin.h1.24_jan_19
Epoch 29/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1155 -
val_loss: 0.3556

Epoch 00029: val_loss improved from 0.36006 to 0.35560, saving model to model
_eng2hin.h1.24_jan_19
Epoch 30/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1105 -
val_loss: 0.3520

Epoch 00030: val_loss improved from 0.35560 to 0.35197, saving model to model
_eng2hin.h1.24_jan_19
Epoch 31/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1063 -
val_loss: 0.3524

Epoch 00031: val_loss did not improve from 0.35197
Epoch 32/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.1023 -
val_loss: 0.3482

Epoch 00032: val_loss improved from 0.35197 to 0.34819, saving model to model
_eng2hin.h1.24_jan_19
Epoch 33/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0998 -
val_loss: 0.3441

Epoch 00033: val_loss improved from 0.34819 to 0.34411, saving model to model
_eng2hin.h1.24_jan_19
Epoch 34/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0967 -
val_loss: 0.3448

Epoch 00034: val_loss did not improve from 0.34411
Epoch 35/50
28125/28125 [==============================] - 168s 6ms/step - loss: 0.0942 -
val_loss: 0.3456

Epoch 00035: val_loss did not improve from 0.34411
Epoch 36/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0934 -
val_loss: 0.3442

Epoch 00036: val_loss did not improve from 0.34411
Epoch 37/50
28125/28125 [==============================] - 171s 6ms/step - loss: 0.0917 -
val_loss: 0.3446

Epoch 00037: val_loss did not improve from 0.34411
Epoch 38/50
28125/28125 [==============================] - 169s 6ms/step - loss: 0.0896 -
val_loss: 0.3428

Epoch 00038: val_loss improved from 0.34411 to 0.34282, saving model to model
_eng2hin.h1.24_jan_19
Epoch 39/50
28125/28125 [==============================] - 170s 6ms/step - loss: 0.0883 -
val_loss: 0.3449

Epoch 00039: val_loss did not improve from 0.34282
Epoch 40/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0873 -
val_loss: 0.3456

Epoch 00040: val_loss did not improve from 0.34282
Epoch 41/50
28125/28125 [==============================] - 168s 6ms/step - loss: 0.0862 -
val_loss: 0.3469

Epoch 00041: val_loss did not improve from 0.34282
Epoch 42/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0855 -
val_loss: 0.3451

Epoch 00042: val_loss did not improve from 0.34282
Epoch 43/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0842 -
val_loss: 0.3472

Epoch 00043: val_loss did not improve from 0.34282
Epoch 44/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0842 -
val_loss: 0.3473

Epoch 00044: val_loss did not improve from 0.34282
Epoch 45/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0832 -
val_loss: 0.3450

Epoch 00045: val_loss did not improve from 0.34282
Epoch 46/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0824 -
val_loss: 0.3463

Epoch 00046: val_loss did not improve from 0.34282
Epoch 47/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0825 -
val_loss: 0.3461

Epoch 00047: val_loss did not improve from 0.34282
Epoch 48/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0819 -
val_loss: 0.3457

Epoch 00048: val_loss did not improve from 0.34282
Epoch 49/50
```

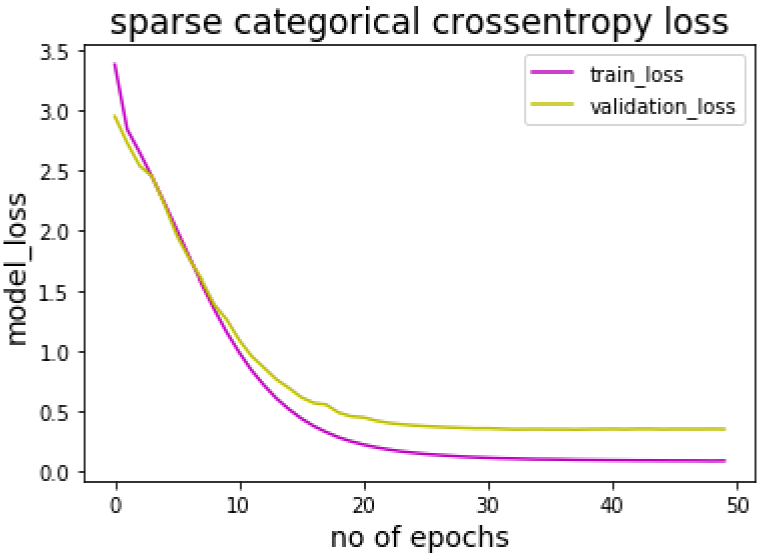
```
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0807 - 
val_loss: 0.3469

Epoch 00049: val_loss did not improve from 0.34282
Epoch 50/50
28125/28125 [==============================] - 167s 6ms/step - loss: 0.0815 - 
val_loss: 0.3457

Epoch 00050: val_loss did not improve from 0.34282
```

In [81]:

```
plt.plot(history.history['loss'],c='m')
plt.plot(history.history['val_loss'],c='y')
plt.legend(['train_loss','validation_loss'])
#ref:https://in.mathworks.com/help/matlab/ref/title.html
plt.title('sparse categorical crossentropy loss',fontsize = 17)
plt.xlabel('no of epochs',fontsize = 14)
plt.ylabel('model_loss',fontsize = 14)
plt.show()
```



In [47]:

```
model = load_model('model_eng2hin.h1.24_jan_19')
preds = model.predict_classes(testX.reshape((testX.shape[0],testX.shape[1])))
```

In [48]:

```
#ref:https://www.analyticsvidhya.com/blog/2019/01/neural-machine-translation-keras/

def get_word(n, tokenizer):
    for word, index in tokenizer.word_index.items():
        if index == n:
            return word
    return None
```

In [49]:

```
preds_text = []
for i in preds:
    temp = []
    for j in range(len(i)):
        t = get_word(i[j], hin_tokenizer)
        if j > 0:
            if (t == get_word(i[j-1], hin_tokenizer)) or (t == None):
                temp.append('')
            else:
                temp.append(t)
        else:
            if(t == None):
                temp.append('')
            else:
                temp.append(t)

    preds_text.append(' '.join(temp))
```

In [57]:

```
pred_df = pd.DataFrame({'actual_words' : test.iloc[:,1], 'predicted_words' : p
reds_text})
pred_df.sample(50)#random words
```

Out[57]:

	actual_words	predicted_words
28344	कलाकारः	कलाकारः
46697	वेबसाइटों खुद के सीएसएस को संशोधित करने के लिए...	कस्टम सीएसएस फ़ाइल का उपयोग करने के लिए
36407	काल इतिहास देखें	काल इतिहास देखें
37173	कृपया अपना कनेक्शन प्रकार चुनें:	कृपया अपना कनेक्शन प्रकार चुनें:
29279	सामान्य	सामान्य
32492	कम्पाइल किया गया	कम्पाइल किया गया
34736	सही	किसी के लिए जांच
47371	दिया सत्र फ़ाइल लोड करें	दिया सत्र फ़ाइल लोड करें
27318	क्या फ़ाइल पूर्वालोकन को दिखाना है इसे प्रयोग ...	इसे प्रयोग के लिए सही पर सेट करें
17115	असफल को मिटाएँ से नहीं से	असफल को मिटाएँ से नहीं से
44437	समर्थित छवि फ़ाइल	समर्थित छवि फ़ाइल
1232	त्रुटि	त्रुटि
44424	छवि डॉटा	छवि डॉटा
21656	पहुँच योग्य वर्णन	पहुँच योग्य वर्णन
811	पहुंचनीय आसंधि नोड को चुनते समय हाइलाइट बक्से ...	नोड को चुनते समय हाइलाइट बक्से की अवधि
36565	काल्स इतिहास	काल्स इतिहास
45040	फ़ॉन्ट्स और रंग	फ़ॉन्ट और को
3231	भिन्न प्रकार के सॉलिटेयर खेल खेलें	भिन्न प्रकार के सॉलिटेयर खेल खेलें
48332	सुरक्षाः	सुरक्षाः
22264	मैसेज नहीं लिख सका	से वापस से
36542	इको निरस्तीकरण सक्रिय करें	इको निरस्तीकरण सक्रिय करें
46072	पूर्ववर्ती विकासकर्ताः	पूर्ववर्ती विकासकर्ताः
38555	ऑफलाइन संपर्क दिखाएँ	कड़ीबद्ध संपर्क
24273	विभाजित करें	विभाजित करें
43264	इस अनुप्रयोग पर मदद	इस अनुप्रयोग पर मदद
4178	एस्लेरियॉट सॉलिटेयर	एस्लेरियॉट सॉलिटेयर
2374	फ्रीसेल सॉलिटेयर	फ्रीसेल सॉलिटेयर
21339	अज्ञात	अज्ञात
45286	पिछला टैब सक्रिय करें	पिछला टैब सक्रिय करें
47773	प्रधान कूटशब्द जरूरी	प्रधान कूटशब्द जरूरी
44858	इस टैब को बन्द करें	इस टैब को बन्द करें
12377	पायथन मददगार	पायथन सहायक
21127	उल्लेखित करता है कि क्या एटीकेहायपरलिंक वस्तु ...	करता है कि क्या एटीकेहायपरलिंक वस्तु चयनित है

	actual_words	predicted_words
30365	अंतिम कोण	अंतिम कोण
13466	रीलोड फ़ाइल	रीलोड फ़ाइल
3529	हुकुम का इक्का	हुकुम का इक्का
10339	को खोलें फ़ाइल से नहीं	को खोलें फ़ाइल से नहीं
42074	कनेक्शन पैरामीटर संपादित करें	कनेक्शन पैरामीटर संपादित करें
1278	मदद	मदद
45208	इतिहास विंडो बंद करें	इतिहास विंडो बंद करें
47284	स्थानीय फ़ाइलें	स्थानीय फ़ाइलें
31619	अंतरण	अंतरण
30265	लागू करने के लिए टिंट	लागू करने के लिए टिंट
27945	डीवीडी में छवि लिख रहा है	डीवीडी में छवि लिख रहा है
3795	मकड़ी	मकड़ी
11499	मिटाया	मिटाया
25846	आकार पा रहा है	आकार पा रहा है
40177	समूह हटा रहा है	समूह हटा रहा है
10026	त्रुटि पर जारी रखें	त्रुटि पर जारी रखें
20665	तत्व से	तत्व से

```
In [ ]:
```

```
In [ ]:
```